# Sarah F. Majors

(724) 683 - 5500                                                                                                    sfmajors373@gmail.com
https://www.linkedin.com/in/sarah-f-majors/                                                    www.github.com/sfmajors373

## OVERVIEW

Originally an archaeologist, I have transitioned to tech and am now looking to obtain a job in artificial intelligence. Currently taking a course on sensor fusion in C++ to further my understanding of the intersection of AI and robotics.

## MACHINE LEARNING PROJECTS

### Artillery Building Damage Detection
**Domains: Computer Vision, Deployment, DevOps, MLOps**
- Deployed an automated machine learning pipeline on AWS using Docker, Nvidia Triton, DVC, MLFlow, git
- Utilized a convolutional neural network to locate building footprints in satellite imagery before damage
- Using the output of the convolutional neural network and a satellite image of the same area but having sustained damage, ResNet50 output a classification for each footprint in the image from not damaged to destroyed
- Incorporated satellite imagery and data from UNOSAT, xBD, and PlanetLabs to achieve a large enough dataset to train, mitigate low resolution images, and access current data
- Created an API using FastAPI allowing users to upload satellite images and receive annotated images with damage information without using the command line or a notebook

### Breast Cancer Detection
**Domains: Computer Vision, Deployment**
- Neural network was designed and trained against stained lymph node slides, from the Camelyon 16 dataset, to perform image segmentation with the goal of the detection of cancerous and abnormal cell structures
- Various methods such as OTSU were used to efficiently clean, tile, and mask the slide data to reduce man hours necessary for production of a dataset as well as enhance the quality of the training set and model accuracy
- Convolutional Neural Network model was designed to flag images from Camelyon 16 dataset as abnormal
- UNet based image segmentation was performed agaisnt Camelyon 16 dataset images identified by the CNN
- Pix2Pix Generative Adversarial Network was used to expand the existing Camelyon 16 dataset to create more training data to feed the pipeline to enhance reliablity and accuracy by balancing positive and negative samples
- Both GPU and CPU computation were utilized to reduce training time and increase the velocity of development
- Set up and deployed a Jupyter Labs based research environment in a Docker container based on Arch Linux on a 32 core Threadripper machine with a mirror of the Camelyon 16 dataset for each team member to reduce the set up time and allow for rapid iterations
- Deployed the model in a flask application on Google Cloud allowing a user to upload a whole slide or just an image tile and see the prediction as well as the segmentation, if applicable

### Privacy Preserving Satellite Imagery
**Domains: Computer Vision, Privacy Enhancing Technologies**
- Produced a demo for Open Mined to implement a machine federated learning model in which data owners have control of the data and the data scientists never access the raw data thereby keeping the data private whilst still allowing the end user to have meaningful results
- The model is trained on satellite imagery to locate pools in certain areas as a proof of concept that these methods of data privacy can be used on data of this nature
- Created reusable Docker container hosting Jupyter Labs and all necessary machine learning libraries to reduce issues related to dependency management
- Compiled PySyft and PyGrid packages for Arch Linux and became the maintainer of those and several other machine learning packages in the Arch User Repository
- Built computer to reduce dependency on expensive cloud compute and prevent runaway costs while experimenting with various models, parameters, etc as well as speed up training time

### Chest X-Ray Pathology Detection
**Domains: Computer Vision, ML Deployment**
- Utilized the CheXpert dataset to develop a potential low cost solution to detect chest pathologies, such as cardiomegaly, edema and pleural effusion, deployed to Raspberry Pi via Flask with three other people
- Trained various models, such as VGG and Resnet, to achieve best results in given time, with VGG16 having the best results at 79.15% accuracy rate with a 50,000 image training set
- Created a small flask application to deploy on Raspberry Pi to upload images and make pathology predictions
- Recorded video demo of the project for hackathon submission

**Federated Learning Cluster**
**Domains: Privacy Enhancing Technologies, Federated Machine Learning, ARM**
- Created a federated learning computational cluster on a set of four Raspberry Pi 3s and 4s to have a platform to further explore how federated learning works
- Built an automated tool to create Arch Linux ARM rootfs for Raspberry Pi 3s and 4s using pacman and bash
- Practiced compiling Arch packages for ARM so I could use Arch on the Pis

**GraphSlam** *Udacity Project*
**Domains: Machine Learning**
- Learn basic 2D Landmark Detection and Tracking
- Practice utilizing Kalman Filters, probability, linear algebra and learn basic motion models

## C++ PROJECTS

**C++ MNIST Neural Net**
**Domain: C++, Computer Vision**
- Utilized Torch and OpenCV to create a neural network to train on the MNIST dataset
- User could upload an appropriately sized image from the command line and the network makes a prediction of the numeral

**C++ htop Replica** *Udacity Project*
**Domain: C++ basics**
- Practiced reading in data and making classes by tracking memory and CPU usage as well as active processes
- Practiced printing data to the terminal and utilizing a premade library to display the data

**C++ Traffic Simulation** *Udacity Project*
**Domain: C++ basics**
- Practiced classes, private and public methods, and various basics to create a very basic traffic simulation
- Guided project with some premade code for visualization

**C++ Memory Management "Chatbot"** *Udacity Project*
**Domain: C++ basics**
- Practiced the Rule of Five, exclusive ownership, and smart pointers to make a mini chatbot
- Guided project with some premade code for visualization

## WORK HISTORY

**Teamsense** *January 2022 - June 2022*
**Software Engineer**
- Tech Stack: Python, Django, Typescript, React, Datadog, git

**Rivers Agile** *April 2019 - January 2022*
**Software Consultant**
- Tech Stack: Python, Plot.ly, Django, RabbitMQ, MQTT, VueJS, node, selenium, i18n, Docker, Vagrant, Gitea

**Contractor** *February 2018 - Janary 2019*
**Data Analyst**
- Tech stack: Python, Jupyter Notebooks, Golang, SQL, Matplotlib

**Nightingale Security** *October 2017 - February 2018*
**Software Intern**
- Tech stack: JavaScript, PHP, Angular, GoogleMapsAPI, git

## EDUCATION

**Fourth Brain** *2020, 2022*
Machine Learning, MLOps

**LambdaSchool** *2017-2018*
Computer Science

**Mercyhurst University**
Bachelors of Science in Anthropology